LIBRARY MANAGEMENT SYSTEM

Aim :

To Implement Library Management System with database using front end tool(Java fx)

Schema :

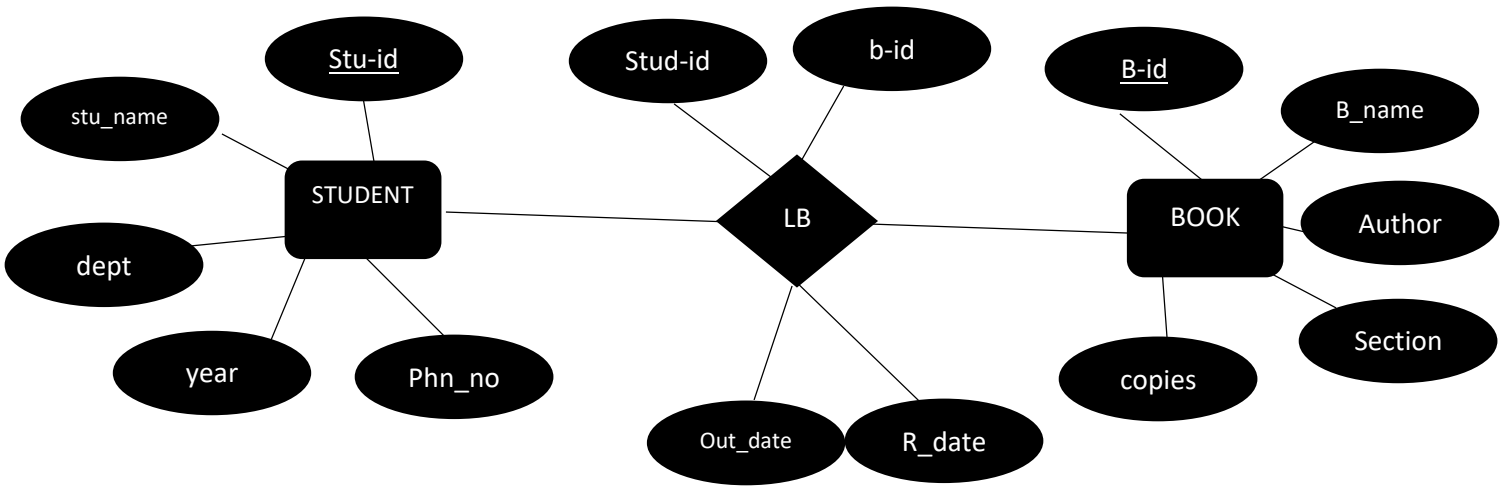
Student(Stu-id , Stu_name , Dept , Year , Phn_no)

Book(B-id , B_name , Author , Section , Copies)

Lb(Stud_id , B_id , out_date , R_date)

All these three tables are in Normalized form.

ER-Diagram :



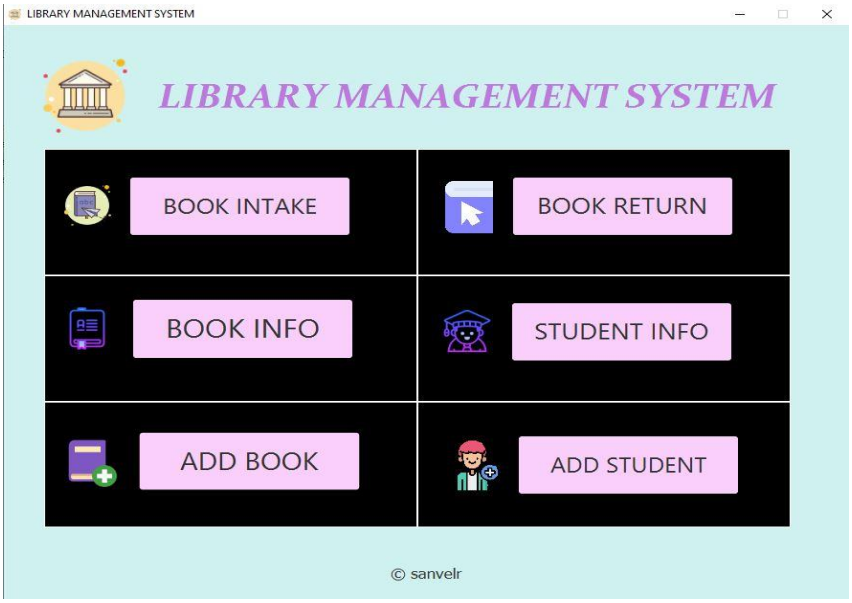
JAVAFX :

- JavaFX is a java library that is used to develop Desktop application as well as Rich Internet Application(RIA).
- Usually it requires three files namely main java file , fxml file , fxml controller java file

JDBC:

- JDBC is a Java API to connect and execute the query with he database
- In order to work with JDBC we need to download ojdbc6 and load it in project
- Usually it requires seven steps
    1. Import statement
    2. Load and Register the driver
    3. Create a connection
    4. Create a statement
    5. Execute the query
    6. Analyze the result
    7. Close all the connections

MAIN MENU :



Main java Source :

```
import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.stage.Stage;

public class LbManagent extends Application
```

```
{
    @Override
    public void start(Stage stage) throws Exception
    {
        Image img = new Image("/IMG/1.png");
        FXMLLoader root =new FXMLLoader();
        root.setLocation(getClass().getResource("FXMLDocument.fxml"));
        Parent root1 =root.load();
        FXMLDocumentController c = root.getController();
        Scene scene = new Scene(root1);
        stage.setTitle("LIBRARY MANAGEMENT SYSTEM");
        stage.setScene(scene);
        stage.getIcons().add(img);
        c.sets(stage);
        stage.setResizable(false);
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

FXML Controller :

```
import java.net.URL;
import java.sql.*;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.image.Image;
```

```
import javafx.stage.Modality;
import javafx.stage.Stage;
public class FXMLDocumentController implements Initializable
{
    Stage C_s;
    Connection con;
    @FXML
    void addS(ActionEvent event) throws Exception
    {
      try{
        FXMLLoader root=new FXMLLoader();
        root.setLocation(getClass().getResource("AddStudent.fxml"));
        Parent root1 =root.load();
        AddStudentController c = root.getController();
        Image img = new Image("/IMG/1.png");
        Stage stage = new Stage();
        stage.initModality(Modality.APPLICATION_MODAL);
        Scene scene = new Scene(root1);
        stage.setTitle("ADD STUDENT");
        stage.setScene(scene);
        stage.getIcons().add(img);
        stage.setResizable(false);
        c.sets(stage);
        stage.show();
       }
        catch(Exception e)
        {
            Alertmsg.error(C_s,e.toString());
        }
    }
      @FXML
    void addB(ActionEvent event) throws Exception
    {
```

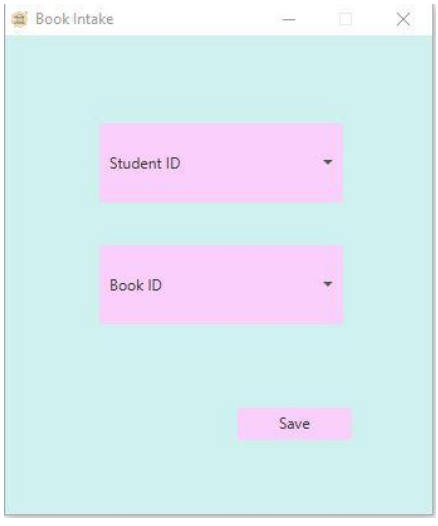
```
   try{
    FXMLLoader root=new FXMLLoader();
    root.setLocation(getClass().getResource("AddBook.fxml"));
    Parent root1 =root.load();
    AddBookController c = root.getController();
    Image img = new Image("/IMG/1.png");
    Stage stage = new Stage();
    stage.initModality(Modality.APPLICATION_MODAL);
    Scene scene = new Scene(root1);
    stage.setTitle("ADD BOOK");
    stage.setScene(scene);
    stage.getIcons().add(img);
    stage.setResizable(false);
    c.sets(stage);
    stage.show();
   }
    catch(Exception e)
    {
        Alertmsg.error(C_s,e.toString());
    }
}
 @FXML
void bookInfo(ActionEvent event)
{
    Stage stage= new Stage();
    stage.initModality(Modality.APPLICATION_MODAL);
    Image img = new Image("/IMG/1.png");
    try
    {
        FXMLLoader root=new FXMLLoader();
        root.setLocation(getClass().getResource("Bookinfo.fxml"));
        Parent root1 =root.load();
        BookinfoController c = root.getController();
```

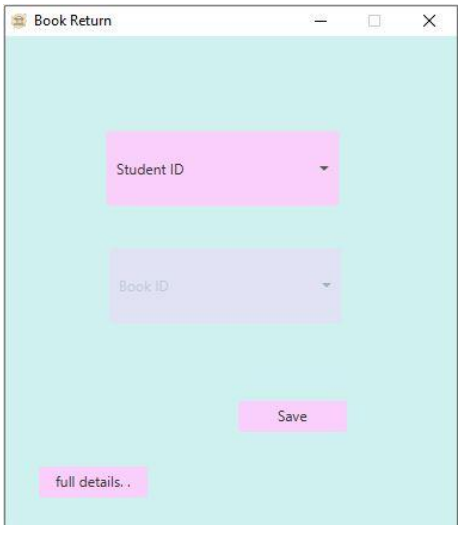
```
        c.sets(stage);
        Scene scene = new Scene(root1);
        stage.setTitle("Book Info");
        stage.setScene(scene);
        stage.getIcons().add(img);
        stage.setResizable(false);
        stage.show();
    }
    catch(Exception e)
    {
         Alertmsg.error(C_s,e.toString());
    }
}
@FXML
void studentInfo(ActionEvent event)
{
    Stage stage= new Stage();
    stage.initModality(Modality.APPLICATION_MODAL);
    Image img = new Image("/IMG/1.png");
    try
    {
        FXMLLoader root=new FXMLLoader();
        root.setLocation(getClass().getResource("Studentinfo.fxml"));
        Parent root1 =root.load();
        StudentinfoController c = root.getController();
        Scene scene = new Scene(root1);
        c.sets(stage);
        stage.setTitle("Student Info");
        stage.setScene(scene);
        stage.getIcons().add(img);
        stage.setResizable(false);
        stage.show();
    } catch(Exception e)
```

```
    {
        Alertmsg.error(C_s,e.toString());
   }}
 @FXML
 void intake(ActionEvent event)
{  try
    {
        Stage stage = new Stage();
        FXMLLoader root =new FXMLLoader();
        root.setLocation(getClass().getResource("BookIntake.fxml"));
        Parent root1 =root.load();
        BookIntakeController c = root.getController();
        Image img = new Image("/IMG/1.png");
        Scene scene = new Scene(root1);
        stage.setTitle("Book Intake");
        stage.setScene(scene);
        stage.getIcons().add(img);
        stage.setResizable(false);
        c.sets(stage);
        stage.show();
    }
    catch(Exception e){
          Alertmsg.error(C_s,e.toString());
     }
 }
 @FXML
 void b_return(ActionEvent event)
 {
   try{
    Stage stage = new Stage();
    FXMLLoader root =new FXMLLoader();
    root.setLocation(getClass().getResource("Return.fxml"));
    Parent root1 =root.load();
```

```
        ReturnController c = root.getController();
        Image img = new Image("/IMG/1.png");
        Scene scene = new Scene(root1);
        stage.setTitle("Book Return");
        stage.setScene(scene);
        stage.getIcons().add(img);
        stage.setResizable(false);
        c.sets(stage);
        stage.show();
      }
     catch(Exception e)
     {
         Alertmsg.error(C_s,e.toString());
     }
    }
    public void sets(Stage s)
    {
       C_s=s;
    }
    @Override
    public void initialize(URL url, ResourceBundle rb) {}
}
```

Book Intake and Return:





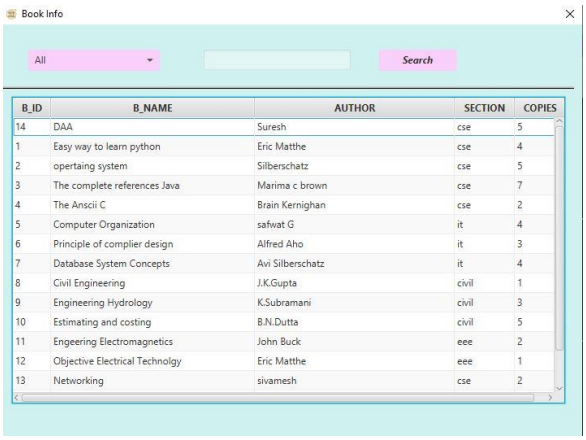
Source:

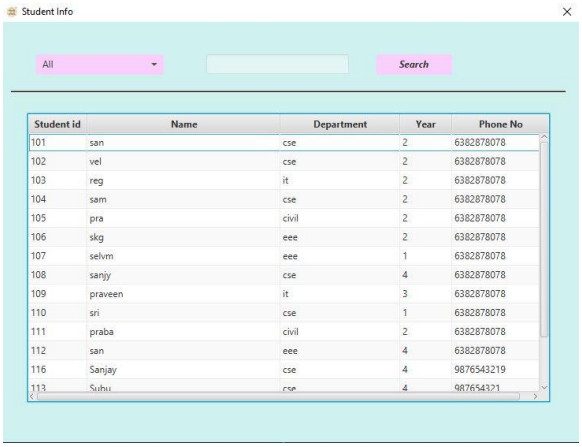
```
import java.sql.*;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.stage.Stage;
public class BookIntakeController implements Initializable {
   Stage C_s;
   @FXML
   private ComboBox<Integer> sid;
   @FXML
   private ComboBox<Integer> bid;
   @FXML
   void save(ActionEvent event)
   {
        if(sid.getSelectionModel().isEmpty() || bid.getSelectionModel().isEmpty())
        {
            Alertmsg.error(C_s, "Select the fields...! ");
            return;
        }
         try
         {
             Class.forName("oracle.jdbc.driver.OracleDriver");
             Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
             PreparedStatement st = con.prepareStatement("insert into lb(stud_id,book_id,out_d) values(?,?,sysdate)");
             st.setInt(1, sid.getValue());
             st.setInt(2, bid.getValue());
             st.execute();
             PreparedStatement st2 = con.prepareStatement("update book set copies_a= copies_a-1 where b_id = ?");
             st2.setInt(1, bid.getValue());
```

```
            st2.execute();
            st.close();
            con.close();
            Alertmsg.message(C_s, "Saved Successfully...! ");
            C_s.close();
         }
        catch(SQLIntegrityConstraintViolationException e)
        {
            Alertmsg.error(C_s, "Book has been already taken.. ");
        }
        catch(Exception e)
        {
           Alertmsg.error(C_s,e.toString());
        }

}
void sets(Stage s)
{
   C_s=s;
}

@Override
public void initialize(URL url, ResourceBundle rb)
{
   try
   {
       Class.forName("oracle.jdbc.driver.OracleDriver");
       Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
       Statement st = con.createStatement();
       ResultSet rs = st.executeQuery("select stu_id from student order by stu_id");
       while(rs.next()){
          sid.getItems().add(rs.getInt(1));
       }
```

```
        ResultSet rs2 = st.executeQuery("select b_id from book where copies_a > 0 order by b_id");
        while(rs2.next()){
            bid.getItems().add(rs2.getInt(1));
        }
        rs.close();
        rs2.close();
        st.close();
    }
    catch(Exception e)
    {     Alertmsg.error(C_s,e.toString());   }       }              }
```

STUDENT AND BOOK INFO :





Source:

```
import java.net.URL;
import java.sql.*;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
```

```
import javafx.stage.Stage;
public class BookinfoController implements Initializable {
    Stage C_s;
    String new_i;
    @FXML
    private TableView<Book> table;
    @FXML
    private TableColumn<Book, Integer> b_id;
    @FXML
    private TableColumn<Book, String> b_name;
    @FXML
    private TableColumn<Book, String> author;
    @FXML
    private TableColumn<Book, String> section;
    @FXML
    private TableColumn<Book, Integer> copies;
    @FXML
    private ComboBox<String> combo;
    @FXML
    private TextField text;
    @FXML
    private Button bt;
    @FXML
    void search(ActionEvent event)
    {
        if(text.getText().equals("") && !(new_i.equals("All")))
        {
            Alertmsg.error(C_s, "Please fill the Search column");
            return;
        }
        table.getItems().clear();
        try
        {
```

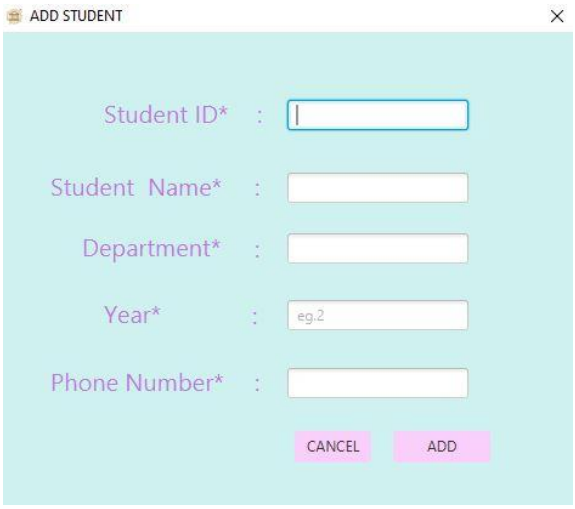
```
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
if(new_i.equals("All"))
{
   table.setItems(getProduct());
}
else if(new_i.equals("B_ID"))
{
   ObservableList<Book> products = FXCollections.observableArrayList();
   PreparedStatement st = con.prepareStatement("select * from book where  b_id = ?");
   st.setInt(1, Integer.valueOf(text.getText()));
   ResultSet rs = st.executeQuery();
   while(rs.next()){
      products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));
   }
   st.close();
   rs.close();
   table.getItems().addAll(products);
 }
else if(new_i.equals("B_Name"))
{
   ObservableList<Book> products = FXCollections.observableArrayList();
   PreparedStatement st = con.prepareStatement("select * from book where  b_name like ?");
   st.setString(1, "%"+text.getText()+"%");
   ResultSet rs = st.executeQuery();
   while(rs.next()){
      products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));
   }
   st.close();
   rs.close();
   table.getItems().addAll(products);
 }
else if(new_i.equals("Author"))
```

```
        {
            ObservableList<Book> products = FXCollections.observableArrayList();
            PreparedStatement st = con.prepareStatement("select * from book where  author like ?");
            st.setString(1, "%"+text.getText()+"%");
            ResultSet rs = st.executeQuery();
            while(rs.next()){
                products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));
            }
            st.close();
            rs.close();
            table.getItems().addAll(products);
         }
        else if(new_i.equals("Section"))
        {
            ObservableList<Book> products = FXCollections.observableArrayList();
            PreparedStatement st = con.prepareStatement("select * from book where  sec = ?");
            st.setString(1, text.getText());
            ResultSet rs = st.executeQuery();
            while(rs.next()){
                products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));
            }
            st.close();
            rs.close();
            table.getItems().addAll(products);
         }

    } catch (Exception ex)
    {
        Alertmsg.error(C_s, ex.toString());
    }
}
@Override
public void initialize(URL url, ResourceBundle rb)
```

```
{
    b_id.setCellValueFactory(new PropertyValueFactory<>("b_id"));
    b_name.setCellValueFactory(new PropertyValueFactory<>("b_name"));
    author.setCellValueFactory(new PropertyValueFactory<>("author"));
    section.setCellValueFactory(new PropertyValueFactory<>("section"));
    copies.setCellValueFactory(new PropertyValueFactory<>("copies"));
    table.setItems(getProduct());
    combo.getItems().addAll("All","B_ID","B_Name","Author","Section");
    combo.getSelectionModel().selectedItemProperty().addListener((v,old,new_i)->
    {
        text.setText("");
        text.setDisable(false);
        bt.setDisable(false);
        if(new_i.equals("All"))
        {
            this.new_i = new_i;
            text.setDisable(true);
        }
        else if(new_i.equals("B_ID"))
            this.new_i = new_i;
        else if(new_i.equals("B_Name"))
            this.new_i = new_i;
        else if(new_i.equals("Author"))
            this.new_i = new_i;
        else if(new_i.equals("Section"))
            this.new_i = new_i;
    });
    combo.getSelectionModel().selectFirst();
}
public void sets(Stage s)
{         C_s = s;     }
public ObservableList<Book> getProduct()
{
```

```
        bt.setDisable(true);
        ObservableList<Book> products = FXCollections.observableArrayList();
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("select * from book");
            while(rs.next()){
                products.add(new Book(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)));
            }
            rs.close();
            st.close();
        } catch (Exception ex)
        {
            Alertmsg.error(C_s, ex.toString());
        }
        return products;
    }
}
```

ADD STUDENT AND BOOK :



Source :

```
import java.net.URL;
import java.sql.*;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.Initializable;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.stage.Stage;
public class AddStudentController implements Initializable
{
    Connection c;
    PreparedStatement ps;
    Stage C_s;
    @FXML
    private TextField t1;
    @FXML
    private TextField t2;
    @FXML
    private TextField t3;
    @FXML
    private TextField t4;
    @FXML
    private TextField t5;
    @FXML
    private Button add;
    @FXML
    private Button cancel;
   @FXML
    void add(ActionEvent event)
    {
```

```
    if(t1.getText().equals("") || t2.getText().equals("") || t3.getText().equals("") || t4.getText().equals("") || t5.getText().equals(""))
    {
        Alertmsg.message(C_s, "Fill all the fields.....");
       return;
    }
    Connection con;
    PreparedStatement ps;
    try{
       Class.forName("oracle.jdbc.driver.OracleDriver");
       con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
       ps = con.prepareStatement("insert into student values(?,?,?,?,?)");
       ps.setInt(1,Integer.valueOf(t1.getText()));
       ps.setString(2,t2.getText());
       ps.setString(3, t3.getText());
       ps.setInt(4, Integer.valueOf(t4.getText()));
       ps.setLong(5, Long.valueOf(t5.getText()));
       ps.execute();
       ps.close();
       con.close();
       Alertmsg.message(C_s, "STUDENT ADDED SUCCESSFULLY.....");
       C_s.close();
    }
    catch(SQLIntegrityConstraintViolationException e)
    {
       Alertmsg.error(C_s," Student Id already Exist..");
    }
    catch(Exception E)
    {
       Alertmsg.error(C_s,E.toString());
    }
  }
  @FXML
  void cancel(ActionEvent event)
```

```
    {
        C_s.close();
    }
    void sets(Stage S)
    {
        C_s = S;
    }
    @Override
    public void initialize(URL url, ResourceBundle rb)
    {
    }
}
```

Result :

Thus the Library Management System using database is implemented Successfully.